**Name : Urvashi Ramdasani**

**Roll No. : 18BCE247**

**Batch : D3**

**Practical No. 1 : Basic Linux Commands**

* **Basic input output operators** :
  + > - It is used for writing output to a file instead of terminal. It always opens file in write mode.

Syntax : Command > File.txt

Example : ./a.out > output.txt

Command 2>&1 - This will display error messages and output on terminal.

Note : The file descriptors -

1. 0 denotes standard input (stdin)
2. 1 denotes standard output (stdout)
3. 2 denotes standard error (stderr)

* >> - This command is also used to store output of a program to a file. It is different from > in the way that >> opens file in append mode.

Syntax : Command >> File.txt

Example : ./a.out > output.txt

* < - It is used for reading input from a file for a program.

Syntax : Command < File.txt

Example : ./a.out < input.txt

* | - It is known as pipe. It is used to direct the output of one program as an input to other program.

Syntax with Example : Suppose we have two files : BasicCProgram.c and InputProgram.c . BasicCProgram.c outputs two values : sum and multiplication of 5 and 2. InputProgram.c requires two inputs and displays their sum as output. If we want to input 7 and 10 (sum and multiplication of 5 and 2 respectively) to InputProgrm.c, we can do the following :

gcc BasicCProgram.c -o obj1

gcc InputProgram.c -o obj2

./obj1 | ./obj2 will output 17 which is the sum of 7 and 10.

* **Basic Linux Commands** :
  1. ls - It lists all the files present inside a directory. It also lists all the directories present in root directory.

b. cd - (Change Directory) It changes the current to the directory specified.

Example - cd Desktop - changes the current working directory to Desktop.

cd .. - it moves to the parent directory of the sub-directory, that is one level up.

c. pwd - (Present Working Directory) It displays the current directory in which we

are working along with the whole path.

d. mkdir - It is used to make directories. Syntax : mkdir \*directoryname\*.

e. rmdir / rm- It is used to remove empty directory.

Syntax : rmdir \*directoryname\* or rm -d \*directoryname\*

To remove directory along with delete all files present inside the directory, use :

Rm -r \*directoryname\*

f. cp - (Copy) It is used to copy the contents of one file to another.

Syntax : cp firstfile secondfile

g. chmod - (Change Mode) It is used to change modes of access of file in a

directory. There are three modes of access : read(r), write(w), execute(x). The

numbers associated with read, write and execute are 4, 2, and 1 respectively. 0

stands for no permission. There are three people categorized in terms of

accessing a file : user(u) that has all permissions, group(g) that has read and

execute permissions and others(o) that can only read.

Syntax : chmod u=rwx, g=rx, o=r \*filename\*

chmod 754 \*filename\* (This gives 7(all permissions) to user, 5(read and

execute) to group and 4(read) to others).

h. chown - (Change Owner) It is used to change the owner of a group or file and give

ownership to other user.

i. chgrp - It is used to change the group ownership of a file or folder.

Syntax with example : sudo addgroup newgroup

sudo chgrp newgroup copy.txt

j. mv - (Move) It is used to move the contents of a file to another file. On moving, the

original file is deleted and a new file is created with the contents of first file.

Syntax : mv \*filename1\* \*filename2\*

k. cat - (Concatenate) It is used to create view copy append multiple files at a time.

Syntax with examples : cat \*filename1\* \*filename2\* - to view two files at time

cat > \*filename\* - creates a file

cat \*filename1\* > \*filename2\* - copies the contents from filename1 to filename2

cat \*filename1\* >> \*filename2\* - appends the contents of filename1 to filename2

l. more / less - When the output of a command is large, more command is used to

view all the outputs at a time. An enter key is used to scroll down to see other

outputs. In less, both the up and down arrow keys can be used to scroll up and

down to view output. It is more efficient than more. Both are implemented using

pipe (|).

Syntax : command | less

m. who / w / whoami - These commands are used to view the users currently working

or the processes running behind.

n. passwd - It is used to change the password of your linux id.

o. find / ff / locate - find searches for a file in a given nested directories. locate is also

used to find the path of a file.

Syntax : find ./urvashi -name copy2.txt

locate copy2.txt

p. du - (Disk Usage) It gives the disk occupied by each file in a directory.

Syntax : du urvashi

du -h urvashi (gives the disk usage in human understandable form like

K,M)

q. sort - It is used to sort the contents of a file. Syntax : sort \*filename\*

r. echo - It is used to display message on terminal. Syntax : echo hello world

s. touch - It is used to create a file from terminal in a directory.

Syntax : touch example.txt

t. quota - It displays user’s disk space and limits.

u. diff - (difference) It is used to find the difference between two files.

Syntax : diff \*filename1\* \*filename2\*

v. cmp - (Compare) It compares two files and tells whether two files are identical or

not. Syntax : cmp \*filename1\* \*filename2\*

w. kill - kill is used to terminate a process manually. Syntax : kill pid (kills the process

with a given pid)

x. ps - It is used to view all the running background processes.

y. gzip - It is used to compress file. Syntax : gzip \*filename\*

z. gunzip - It is used to decompress files. Syntax : gunzip \*filename\*

* **Basic Filter Commands** :

1. df - (Disk Free) It is also used to view the disk used by different files. du gives the disk usage of current directory only but df gives the overall disk usage.
2. grep - It searches the file for a pattern and prints all the lines having that pattern. Syntax : grep \*word-to-be-searched\* \*filename\*
3. head - It prints the line from the beginning of the file. Syntax : head \*filename\*
4. tail - It prints the last 10 lines from the end of the file. Syntax : tail \*filename\*

NOTE: Both head and tail can print any specified number of lines using head -n \*filename\*.

1. wc - (Word Count) It gives the word count in a file. Syntax : wc -w \*filename\*

* **Relative and Absolute Path** :
* **Break Operations** :
  + ctrl + c = It completely terminates the process.
  + ctrl + z = It keeps the process on background, where it is still running. It does not frees the resources, it only frees the terminal.